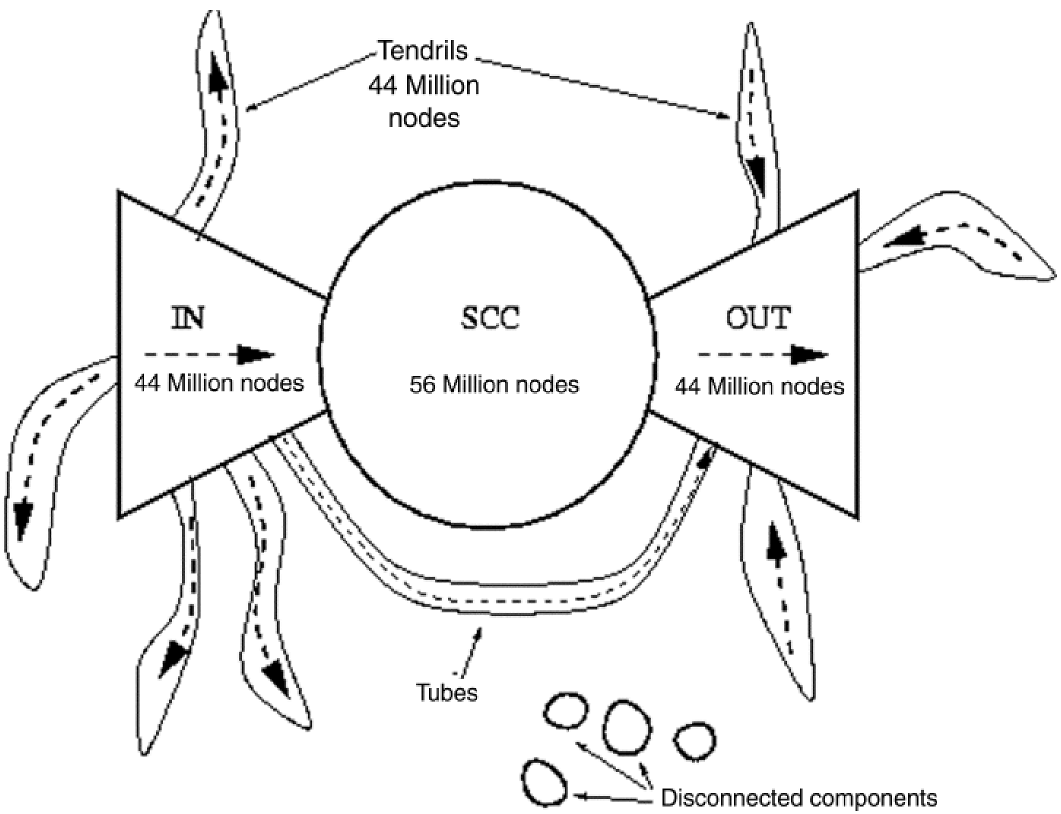
LINK ANALYSIS

Outline

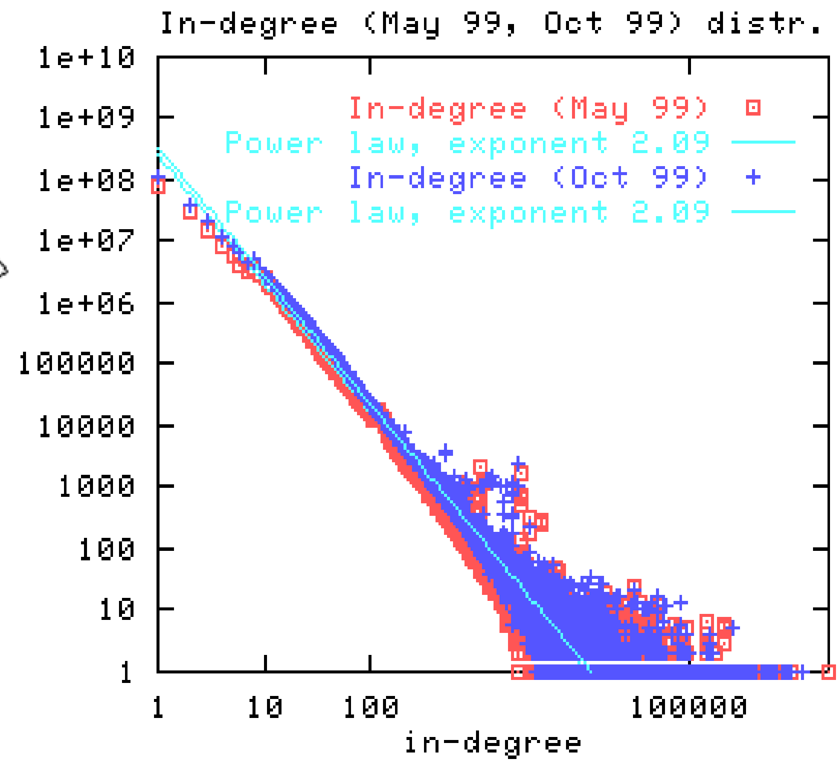
- Intro
- Basics of probability and information theory
- Retrieval models
- Retrieval evaluation
- Link analysis
  - Models for the web graph
  - HITS
  - PageRank
  - SALSA
- From queries to top-k results
- Social search

The web graph

- Study of Broder et al.: *Graph Structure in the Web*. WWW 2000



Web graph structure



In-degree distribution (similar distribution for out-degree)

Possible models to explain the web graph

- **Random graph** (i.e., Erdös-Renyi model)
  - An edge between any two nodes is included with probability, independent of other edges

- **Preferential attachment** (Barabási & Albert; Science 1999)
  - New nodes in the network join with high probability those nodes that already have a high (in)degree (“rich get richer”)

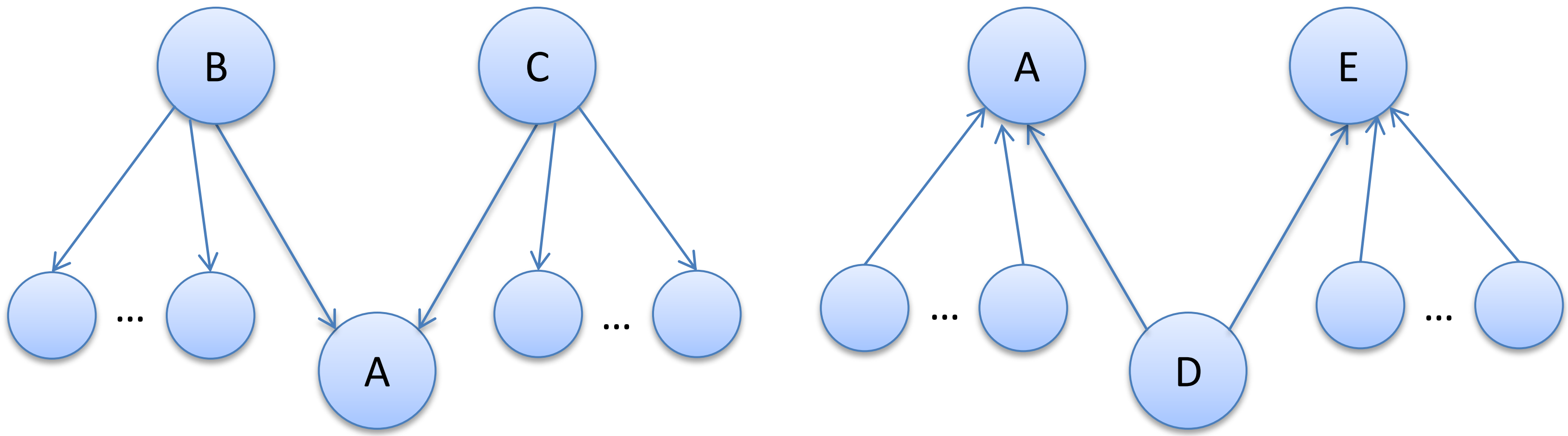
  - Specifically: new node x with m links, chooses node y to link to with probability proportional to $degree(y)$

$$P(x \rightarrow y) = \frac{degree(y)}{\sum_{y'} degree(y')}$$

- **Link copying & preferential attachment**
  - New nodes copy the links of an existing nodes with probability α or follow preferential attachment with probability $(1-\alpha)$

Node authority

- How to measure the authority of a node in the web graph?
- What about citation measures in academic work?
  - E.g., $ImpactFactor(J, Y)$: average number of references from articles published in journal J in year Y to articles published in J in year $Y-1$ and $Y-2$ (not suitable for cross-journal references)
  - Other measures consider following important structures in the citation graph: co-citations & co-referrals



Publication A gets higher authority

D becomes a better “hub”

Web page links vs. citations

- Similarity between web graph and citation graph
  - Citation of related work y by publication x increases the importance of y (i.e., x “endorses” y); links on the web can be viewed as **endorsements** as well.

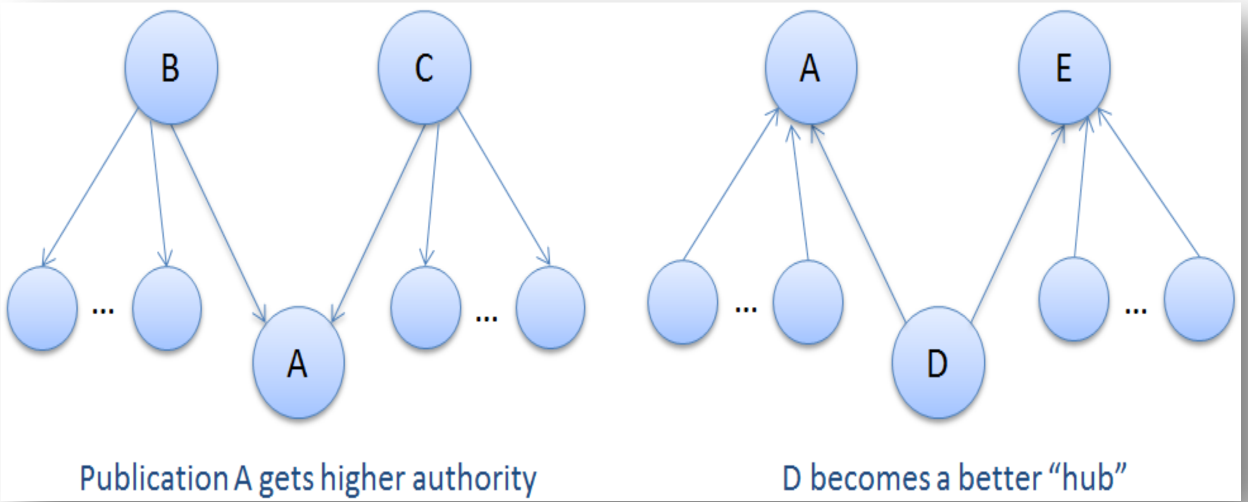
- Differences
  - Web pages with high in- or out-degree could be portals or spam.
  - Company websites don’t point to their competitors.
  - …

Hypertext induced topic selection (HITS)

- By John Kleinberg in 1999

- Every node v has
  - Authority score $A(v)$
  - Hubness score $H(v)$

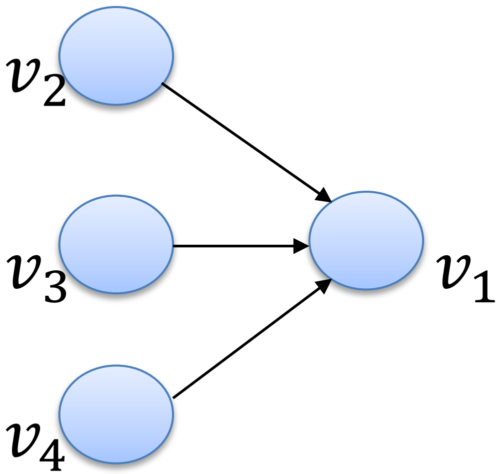
- Example
  - If A, E already have high authority, D becomes a better hub.
  - If B and C are good hubs, A gets higher authority.

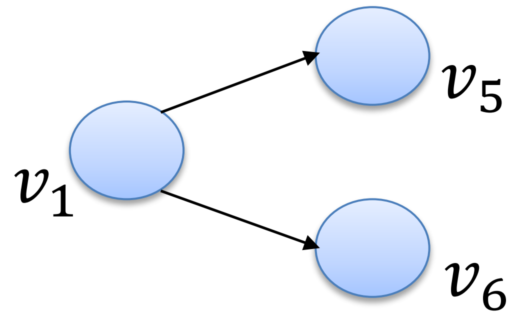


- Web page has high authority score if it has many links from nodes with high hubness scores

- Web page has high hubness score if it has many links to nodes with high authority scores

Computing authority and hubness scores (1)





$a(v_1) = h(v_2) + h(v_3) + h(v_4)$

$h(v_1) = a(v_5) + a(v_6)$

- Compute scores recursively

$$A(v) = \sum_{u \to v} H(u) \qquad H(v) = \sum_{v \to w} A(w)$$

Computing authority and hubness scores (2)

- Let $\vec{a} = \begin{pmatrix} a(v_1) \\ a(v_2) \\ \cdot \\ \cdot \\ \cdot \\ a(v_n) \end{pmatrix}, \vec{h} = \begin{pmatrix} h(v_1) \\ h(v_2) \\ \cdot \\ \cdot \\ \cdot \\ h(v_n) \end{pmatrix}$

- We can write $\vec{a} = M^T\vec{h}$ and $\vec{h} = M\vec{a}$, where M is the adjacency matrix $m_{ij} = 1 \Leftrightarrow v_i \rightarrow v_j$

- By substitution

$$\vec{a} = M^T M\vec{a}$$
$$\vec{h} = MM^T\vec{h}$$

- Interpretation

  $M^T M_{(ij)}$: number of nodes pointing to both i and j

  $MM^T{}_{(ij)}$: number of nodes to which both i and j point

Eigenvectors and Eigenvalues

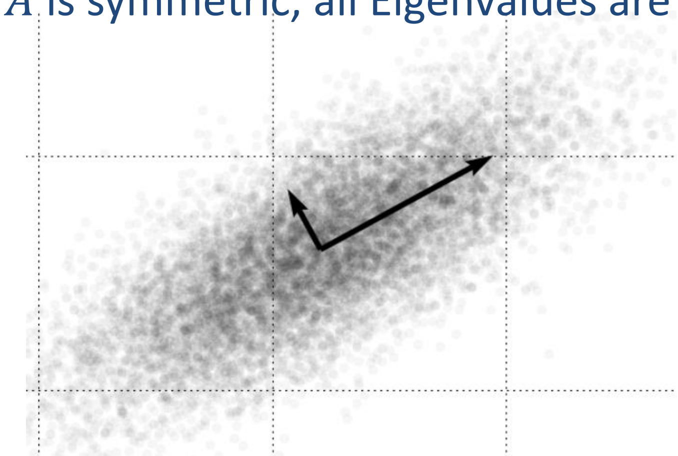
- **Eigenvector, Eigenvalue:** for an $n \times n$ matrix A, $n \times 1$ vector $\mathbf{v}$ and a scalar λ that satisfy $A\mathbf{v} = \lambda\mathbf{v}$ are called Eigenvector and Eigenvalue of A.
  - Eigenvalues are the roots of the **characteristic function**

$$f(\lambda) = \det(A - \lambda I)$$

$$\det(A) = \sum_{\substack{i=1 \\ (j=1)}}^{n} (-1)^{(i+j)} a_{ij} \det(A^{(\setminus \text{ij})})$$

$\boxed{A^{(\setminus \text{ij})} \text{ is } A \text{ without the } i\text{'th row and the } j\text{'th column}}$

  - If A is symmetric, all Eigenvalues are real!



- **Principal eigenvector** gives the direction of **highest variability!**

Authority and hubness scores revisited

$$\vec{a} = M^T M \vec{a}$$

Can be quite dense…

$$\vec{h} = M M^T \vec{h}$$

Eigenvector of $M^T M$

Eigenvector of $M M^T$

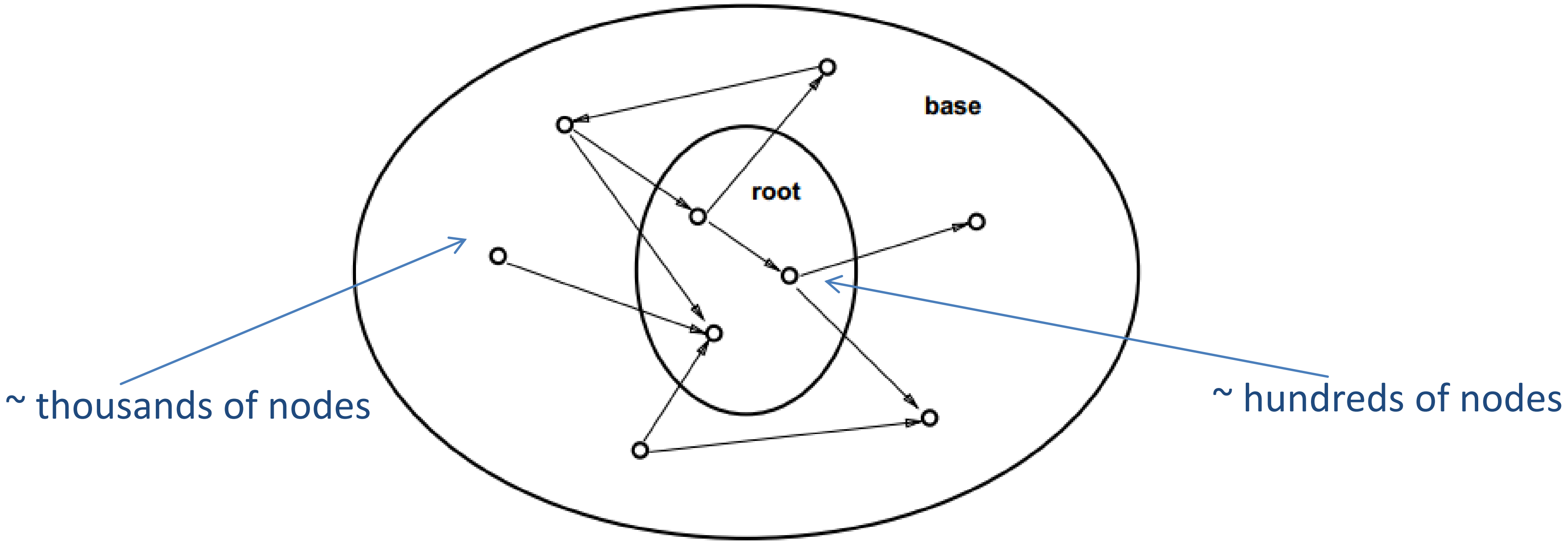
- Algorithm

Start with $\vec{a}^{(0)} = \begin{pmatrix} 1/n \\ 1/n \\ \cdot \\ \cdot \\ \cdot \\ 1/n \end{pmatrix}$, $\vec{h}^{(0)} = \begin{pmatrix} 1/n \\ 1/n \\ \cdot \\ \cdot \\ \cdot \\ 1/n \end{pmatrix}$, where n is the number of nodes

Repeat $\vec{a}^{(i+1)} = M^T \vec{h}^{(i)}$, $\vec{h}^{(i+1)} = M \vec{a}^{(i)}$, L1-normalize $\vec{a}^{(i+1)}$, $\vec{h}^{(i+1)}$ until convergence
(Note: algorithm returns principal Eigenvectors of $M^T M$ and $M M^T$)

The HITS algorithm

1. Start with a "**root set**" of pages relevant to a topic (or information need)
2. Create a "**base subgraph**" by adding to the root set
   – All pages that have incoming links from pages in the root set
   – For each page in the root set: up to k pages pointing to it
3. Compute authority and hubness scores for all pages in the base subgraph
4. Rank pages by decreasing authority scores



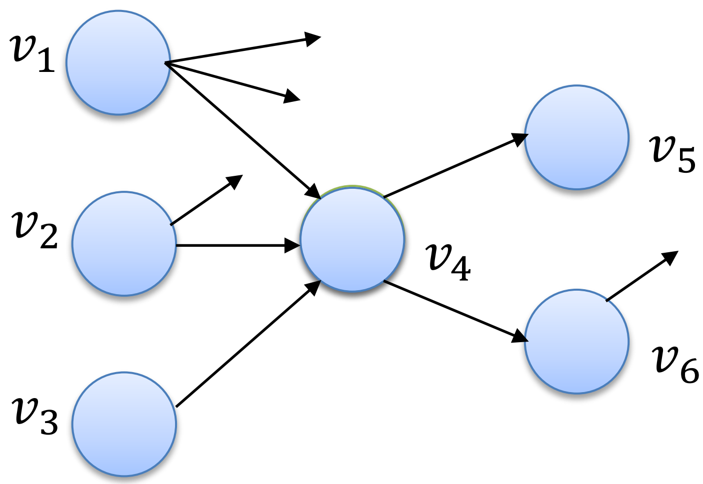
Source: http://www.cs.cornell.edu/home/kleinber/auth.pdf

Drawbacks of HITS

- Relevance of documents in root set is not addressed
- Documents may contain multiple “identical links” to the same document (in some other host)
  → Link spamming is a problem
- Bias towards bipartite subgraphs
- Danger: topic drift – found pages may not be related to the original query
- HITS scores need to be computed at query time (i.e., on the base set)
  → Too expensive in most query scenarios
- Rank is not stable to graph perturbations

Let's think again…

- Incoming links reflect "endorsements" (authority ↗)
- Outgoing links reflect "outflowing" authority (authority ↘)



- Probabilistic view: what is the probability of being at node v_i?

$$P(v_i) = \sum_{v_j} P(v_i|v_j)\, P(v_j)$$

$$P(v_i) = \sum_{v_j \to v_i} P(v_i|v_j)\, P(v_j)$$

Restrict to direct predecessors only!

(Assuming uniform probability of choosing successor: $P(v_i|v_j) = \frac{1}{Outdegree(v_j)}$)

Markov chains

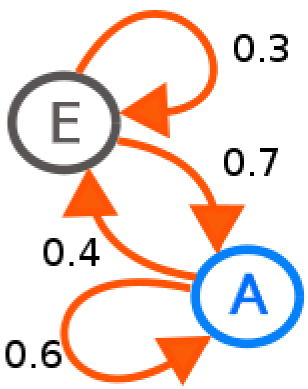
- Chain of discrete random variables

$$X_1 \rightarrow X_2 \rightarrow \quad \cdots \quad \rightarrow X_{n-1} \rightarrow X_n \rightarrow$$

- Markov assumption: a variable is independent of all its non-descendants given its parents

$$P(X_i \mid X_1, X_2, \dots, X_{i-1}) = P(X_i \mid X_{i-1})$$

- Example



States with transition probabilities
Source: Wikipedia

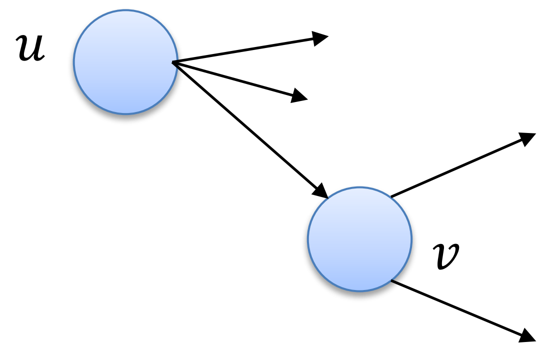
$$X_1 = \mathrm{E} \rightarrow X_2 = \mathrm{E} \rightarrow X_3 = \mathrm{A}$$

Possible instantiation of Markov chain:

$$P(X_i = \mathrm{E}) = P(X_{i-1} = \mathrm{E})P(X_i = \mathrm{E} | X_{i-1} = \mathrm{E}) + P(X_{i-1} = \mathrm{A})P(X_i = \mathrm{E} | X_{i-1} = \mathrm{A})$$

e.g., with $P(X_1 = \mathrm{E}) = P(X_1 = \mathrm{A}) = \frac{1}{2}$

Markov chain model of walking on the web graph



- Stationary probability at node v

$$P(X_i = v) = \sum_{\substack{X_{i-1}=u \\ \wedge\, u \to v}} P(X_{i-1} = u) P(X_i = v | X_{i-1} = u)$$

- Uniform transition probability accros successors

$$P(X_i = v | X_{i-1} = u) = \frac{1}{Outdegree(u)}$$

Markov chain properties (1)

- Homogeneous
  - $P(X_i = v | X_{i-1} = u)$ are independent of i.

- Irreducible
  - Every state is reachable from any other state (with probability>0).

- Aperiodic
  - The **greatest common divisor** of all (recurrence) values l with
    $$P(X_l = v \wedge X_k \neq v \text{ for } k = 1,2, \dots, l-1 | X_0 = v) > 0$$
    is 1 for every v.

- Positive recurrent
  - For every state, the expected number of steps after which it will be reached is finite.

Markov chain properties (2)

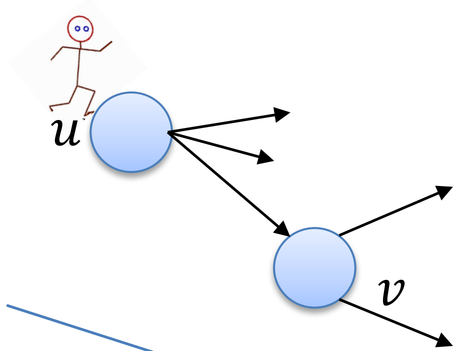
- Ergodic
  - homogeneous, irreducible, aperiodic, and positive recurrent

- Theorem 1
  - A finite-state irreducible Markov chain is ergodic if it has an aperiodic state.

- Theorem 2
  - For every ergodic Markov chain there exist stationary state probabilities.

- Goal: Markov-chain model to compute stationary probabilities for the nodes in the web graph
  - Finite number of nodes (i.e., finite number of states)
  - Not irreducible (i.e., not every node can be reached from every node)
  - States need to be aperiodic

PageRank: a random walk model on the web graph



Uniform random jump probability, i.e., $rj(v) = \frac{1}{n}$, n: number of nodes

Uniform transition probability, i.e., $P(v|u) = \frac{1}{Outdegree(u)}$

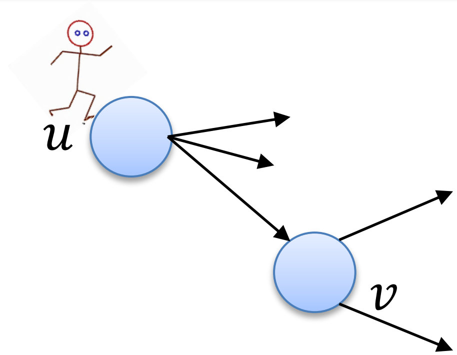
$$P(v) = (1-\alpha) \cdot rj(v) + \alpha \sum_{u \to v} P(u) \cdot P(v|u)$$

- Random walker reaches v by
  - following one of the outgoing links of the predecessors of v with probability α
  - or by randomly jumping to v with probability $1-\alpha$ (**random jump probability**)

- Notes
  1. Every node (i.e., state) can be reached from every other node (through random jumps).
  2. Because of transition cycles of length 1 (from one node to the same node with probability > 0) every node (i.e., state) is aperiodic.

     → Valid ergodic Markov chain model (i.e., with existing **stationary probabilities** for each state)

Computing PageRank scores



$$Authority(v) = PageRank(v) = P(v) = (1-\alpha)\cdot rj(v) + \alpha \sum_{u \to v} P(u)\cdot P(v|u)$$

- Jacobi power iteration method
  - Initalize $\vec{p}^{(0)}, \vec{r}$ with $\vec{p}^{(0)} = \overrightarrow{rj} = \begin{pmatrix} 1/n \\ 1/n \\ . \\ . \\ . \\ 1/n \end{pmatrix}$
  - Repeat until convergence $\vec{p}^{(i+1)} = \alpha M \vec{p}^{(i)} + (1-\alpha)\overrightarrow{rj}$

    where M is transition matrix with $m_{v,u} = \begin{cases} P(v|u) = \frac{1}{Outdegree(u)}, & \text{if } u \to v \\ 0, & \text{otherwise} \end{cases}$

Remarks to Jacobi power iteration for PageRank

$$\vec{p}^{(i+1)} = \alpha M \vec{p}^{(i)} + (1-\alpha)\vec{r}$$

- M is a stochastic matrix (i.e., columns sum up to 1)

- Theorem 3
  - For every stochastic matrix M, all Eigenvalues λ have the property $|\lambda| \leq 1$ and there is an Eigenvector x with Eigenvalue $\lambda_1 = 1$, such that $x \geq 0$ and $\|x\| = 1$.

    x is called the Principal Eigenvector.

- Theorem 4
  - Power iteration converges to principal Eigenvector with convergence rate $\alpha = |\lambda_2/\lambda_1|$, where λ_2 is the second largest Eigenvalue.

Topic-sensitive and personalized PageRank

$$PageRank(v) = P(v) = (1-\alpha) \cdot rj(v) + \alpha \sum_{u \to v} P(u) \cdot P(v|u)$$

- Bias random walk towards pages of certain topic or pages liked by the user

- How could this be done?

  - Possibility 1: introduce classification process into random walk, e.g., page v could be visited with probability proportional to linear combination of $P(v)$ and $P(v|T)$ → difficult to scale

  - Possibility 2: bias random jump towards the set T of target pages (much simpler)

$$rj(v) = \begin{cases} \frac{1}{|T|}, & \text{if } v \in T \\ 0, & \text{otherwise} \end{cases}$$

and run Jacobi iterations for

$$\vec{p}^{(i+1)} = \alpha M \vec{p}^{(i)} + (1-\alpha)\overrightarrow{rj}$$

Topic-sensitive PageRank

- Algorithm by Haveliwala, TKDE 2003
  1. Given multiple classes, precompute for each class T_k a topic sensitive PageRank vector p_k through Jacobi iterations

$$\overrightarrow{p_k}^{(i+1)} = \alpha M \overrightarrow{p_k}^{(i)} + (1-\alpha)\overrightarrow{rj_k}$$

  2. For the user query q compute $P(T_k|q)$
  3. Compute authority score of a page v as

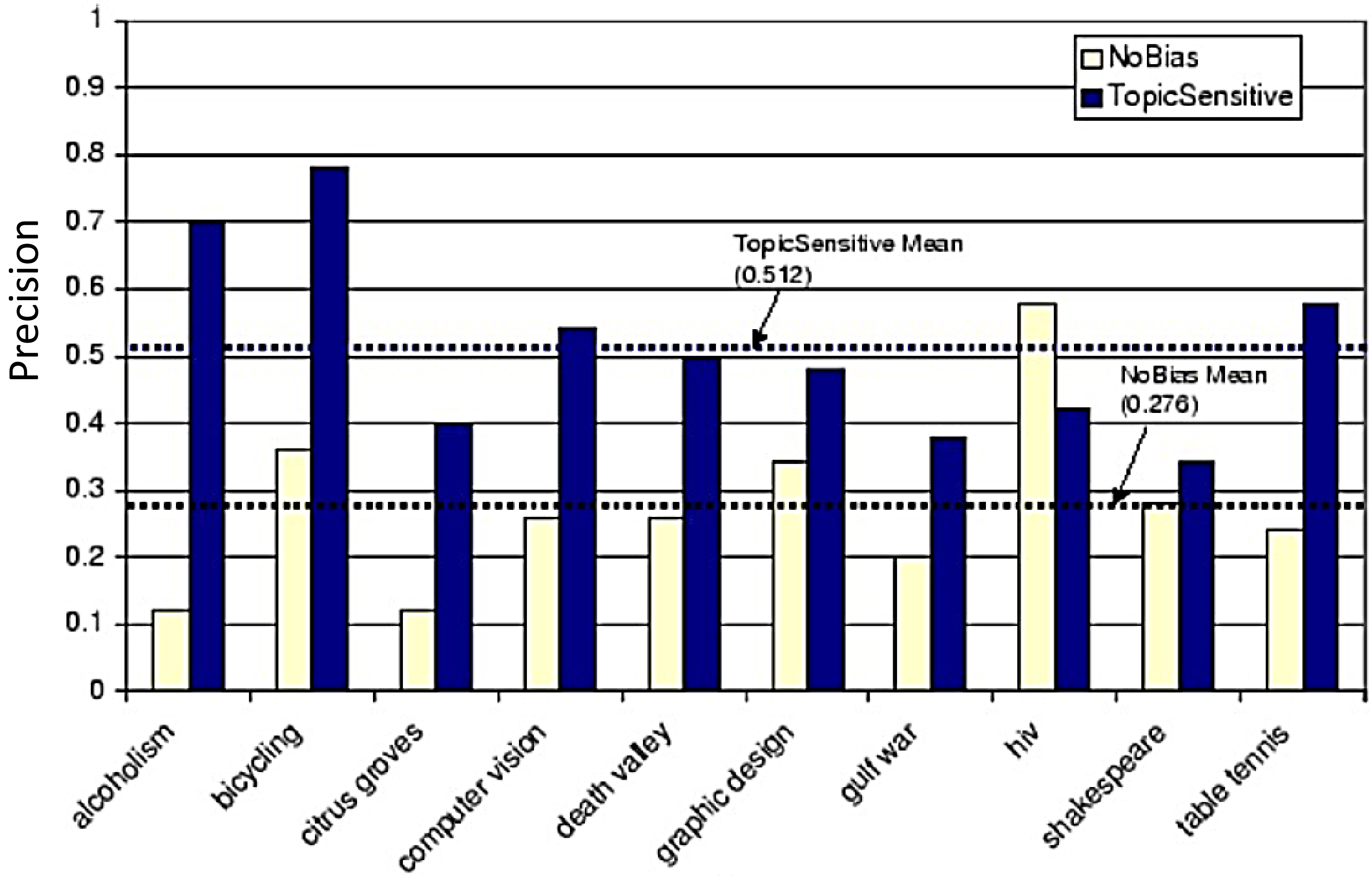
$$\sum_k P(T_k|q) p_k(v)$$

- **Theorem 3**

Let $\overrightarrow{rj_1}$ and $\overrightarrow{rj_2}$ be biased random-jump vectors and $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$ denote the corresponding biased PageRank vectors. For all $\beta_1, \beta_2 \geq 0$ with $\beta_1 + \beta_2 = 1$ it holds:

$$\vec{p} = \beta_1\overrightarrow{p_1} + \beta_2\overrightarrow{p_2} = \alpha M(\beta_1\overrightarrow{p_1} + \beta_2\overrightarrow{p_2}) + (1-\alpha)\left(\beta_1\overrightarrow{rj_1} + \beta_2\overrightarrow{rj_2}\right).$$

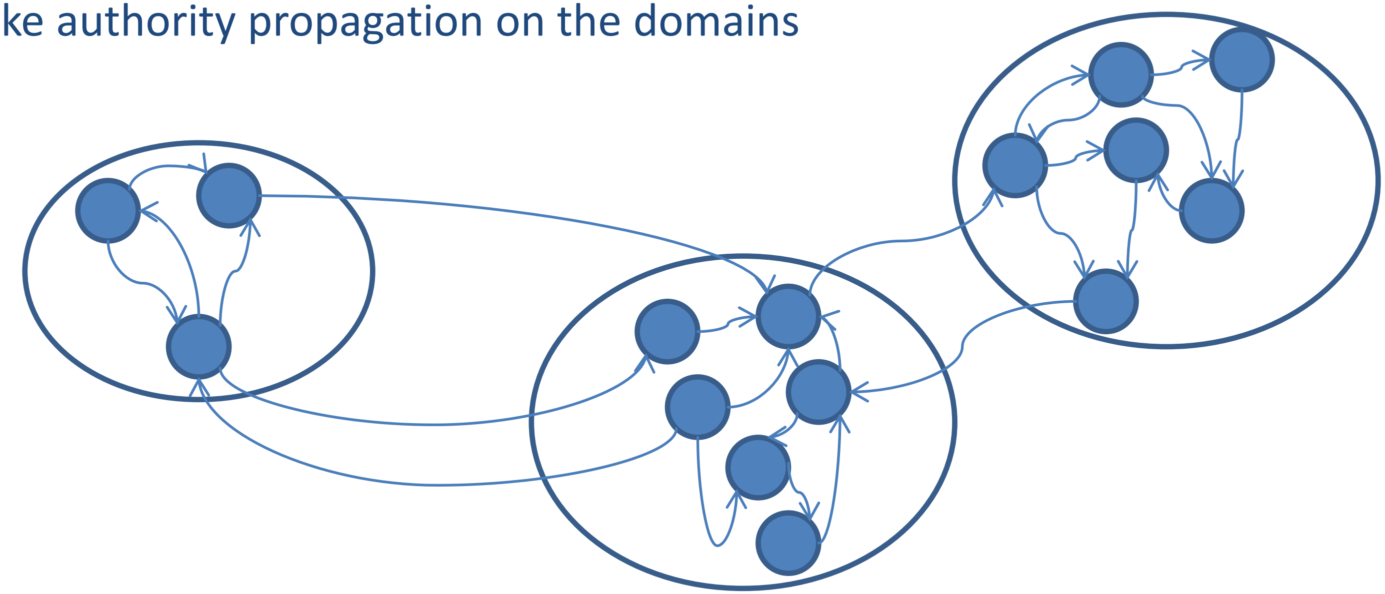
Topic-sensitive PageRank evaluation



Source: Haveliwala,
[Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search]().
TKDE 2003

Efficient PageRank

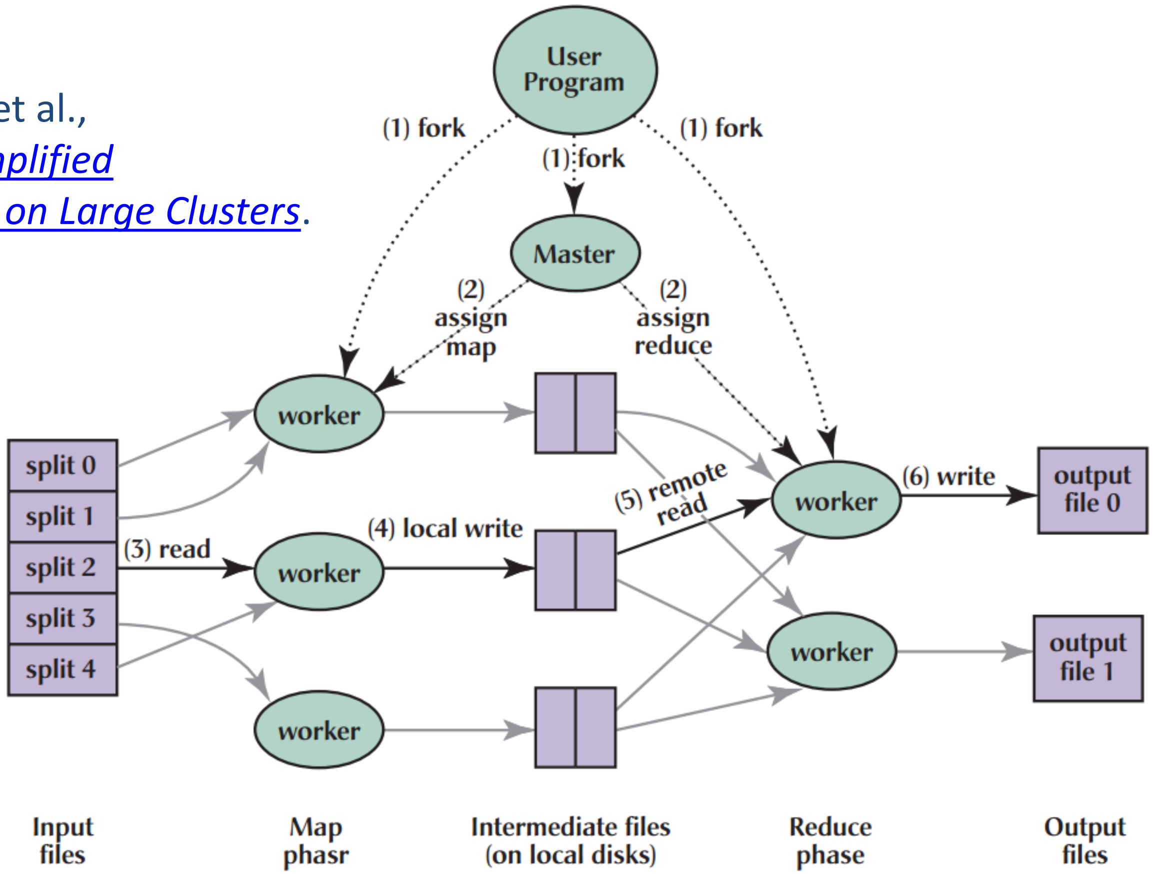
- Distributed PageRank computation
  - Compute PageRank for each page on a domain and combine it with PageRank-like authority propagation on the domains



  - Compute PageRank by using MapReduce
    - Map: $(k1, val1) \rightarrow list(k2, val2)$ *// list of key-value pairs from one domain mapped to list of key-value pairs of another domain*
    - Reduce: $(k2, list(val2)) \rightarrow list(val3)$ *//compute results from each group in parallel*

General framework for MapReduce

Source: J. Dean et al., *MapReduce: Simplified Data Processing on Large Clusters*. CACM 2008



- MapReduce implementations: PIG (Yahoo), Hadoop (Apache), DryadLinq (Microsoft)

MapReduce iteration for PageRank

- **Initial MAP**:

  (url, content)

  ⬇

  {(url, (PR_0, list(urls pointed to by url)))} *//set initial PageRank value*

- **Initial REDUCE**: return tuples unchanged

- **MAP**:

  (url, (PR(url), list(n urls pointed to by url)))

  ⬇

  {(linked url1, PR(url) /n), …,(linked url$n$, PR(url) /n), (url, list(n urls pointed to by url))}

- **REDUCE**:

  (url, {(url', PR')|url' points to url}U {(url, list(urls pointing to url)})

  ⬇

  {(url, (PR(url), list(urls pointed to by url)))}

PageRank: strengths & weaknesses

- Strengths
  - Elegant theoretical foundation with nice interpretations (e.g., stochastic matrices, principal Eigenvectors, stationary state probabilities in random walk model/ Markov-chain process)
  - Can be extended to topic-based and personalized models
  - Static measure (i.e., can be precomputed)
  - Relatively straight-forward to scale
  - PageRank scores are relatively stable to graph perturbations

- Weaknesses
  - Rank is not stable to perturbations
  - Random-jump extension is needed to make corresponding Markov process ergodic on arbitrary directed graphs
  - Query independence (no direct relation to relevance)

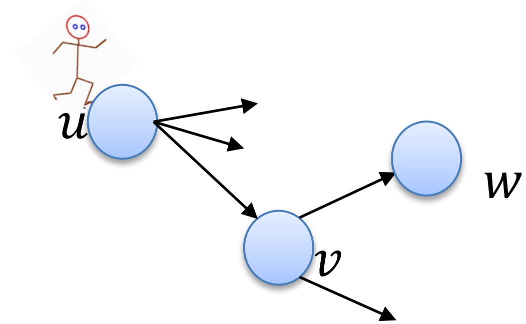
SALSA: stochastic approach for link-structure analysis

- 2-hop random walk model for HITS on connected directed graphs

  Likelihood of reaching w from u in two hops is

  $$\sum_{v \in V: u \to v \to w} \frac{1}{Outdegree(u)} \frac{1}{Outdegree(v)}$$

  (assuming uniform probability of choosing successor)



- From original graph $G(V, E)$, construct undirected bipartite graph $G'(V', E')$:

  $V' = \{u_h | u \in V \wedge Outdegree(u) > 0\}$

  $\cup \{u_a | u \in V \wedge Indegree(u) > 0\}$ and

  $E' = \{\{u_h, v_a\} | u \to v \in E\}$.

- Construct from G' hub matrix H: $h_{uw} = \sum_{v \in V: u_h - v_a - w_h} \frac{1}{Degree(u)} \frac{1}{Degree(v)}$
- Construct from G' auth. matrix A: $a_{uw} = \sum_{v \in V: u_a - v_h - w_a} \frac{1}{Degree(u)} \frac{1}{Degree(v)}$
- Can be extended with random-jump probabilities

SALSA: strengths and weaknesses

- Strengths
  - Probabilistic HITS-based (i.e., random-walk) interpretation
  - Matrices H and A are stochastic → principal eigenvectors hold stationary state probabilities
- Weaknesses
  - Random-jump extension is definitely needed to make corresponding Markov process ergodic on arbitrary directed graphs
  - Authority scores are proportional to indegrees of nodes (in the original graph); the same holds for hubness scores and outdegrees

    → suggests much simpler ranking, e.g., based on in-degrees only

Precision@k ranking comparison

Source:
Borodin et al.,
Link analysis ranking: algorithms, theory, and experiments.
TOIT 2005

Query	HITS	PAGERANK	INDEGREE	SALSA	HUBAVG	MAX	AT-MED	AT-AVG	BFS	BAYESIAN	SBAYESIAN
abortion	90%	70%	100%	100%	100%	100%	100%	100%	100%	90%	100%
affirmative action	70%	50%	50%	50%	10%	10%	10%	10%	80%	40%	50%
alcohol	90%	60%	90%	90%	90%	80%	80%	80%	90%	80%	90%
amusement parks	100%	30%	30%	50%	0%	90%	10%	0%	80%	100%	30%
architecture	10%	70%	70%	70%	10%	60%	70%	10%	60%	10%	70%
armstrong	20%	50%	20%	20%	20%	20%	20%	20%	50%	20%	20%
automobile industries	10%	10%	20%	30%	10%	10%	10%	10%	60%	20%	20%
basketball	0%	70%	20%	20%	0%	10%	10%	10%	100%	10%	20%
blues	60%	80%	60%	60%	70%	60%	70%	70%	50%	60%	60%
cheese	0%	20%	30%	30%	10%	0%	0%	10%	50%	0%	30%
classical guitar	90%	50%	70%	70%	50%	80%	50%	50%	90%	70%	70%
complexity	0%	50%	50%	50%	0%	90%	90%	0%	80%	0%	50%
computational complexity	90%	70%	90%	90%	90%	90%	90%	90%	90%	100%	90%
computational geometry	100%	40%	70%	70%	70%	100%	70%	70%	100%	80%	70%
death penalty	100%	70%	90%	90%	70%	100%	100%	100%	100%	100%	90%
genetic	100%	70%	100%	100%	100%	100%	100%	100%	90%	100%	100%
geometry	90%	20%	90%	90%	90%	90%	90%	80%	90%	80%	90%
globalization	100%	70%	90%	90%	100%	100%	100%	100%	90%	100%	90%
gun control	0%	50%	100%	100%	100%	100%	100%	100%	100%	80%	100%
iraq war	40%	30%	30%	30%	10%	20%	20%	10%	90%	40%	30%
jaguar	0%	30%	0%	0%	0%	0%	0%	0%	10%	0%	0%
jordan	0%	30%	30%	30%	40%	100%	100%	100%	40%	30%	30%
moon landing	0%	30%	20%	20%	0%	0%	0%	0%	100%	0%	20%
movies	10%	20%	50%	40%	50%	70%	70%	70%	60%	60%	50%
national parks	0%	50%	10%	10%	80%	80%	80%	0%	70%	0%	10%
net censorship	0%	30%	80%	80%	60%	90%	90%	90%	80%	70%	80%
randomized algorithms	70%	80%	80%	80%	40%	50%	50%	50%	60%	80%	70%
recipes	0%	20%	70%	70%	30%	90%	90%	100%	80%	80%	70%
roswell	0%	20%	40%	40%	70%	70%	60%	0%	60%	10%	40%
search engines	80%	90%	100%	100%	100%	100%	100%	100%	90%	80%	100%
shakespeare	100%	70%	100%	100%	100%	100%	100%	100%	100%	100%	100%
table tennis	90%	60%	100%	100%	90%	90%	90%	90%	90%	90%	100%
weather	80%	50%	80%	80%	60%	80%	80%	80%	90%	80%	80%
vintage cars	20%	10%	60%	60%	20%	60%	60%	20%	70%	40%	60%
avg	47%	48%	61%	62%	51%	67%	64%	54%	78%	56%	61%
max	100%	90%	100%	100%	100%	100%	100%	100%	100%	100%	100%
min	0%	10%	0%	0%	0%	0%	0%	0%	10%	0%	0%
stdev	43%	23%	31%	31%	38%	36%	36%	42%	21%	37%	31%

Summary

- Web links vs. citation Links
- Hubs and Authorities → HITS
- Random walks on the web graph → PageRank
  - Topic-sensitve, personalized
  - Distributed
- Two-hop random walks between hubs and authorities → SALSA